Игра «Звездный бой» для домашних компьютеров

А. К. ДЬЮДНИ

В НАЧАЛЕ 70-х годов я часто засиживался до позднего вечера в своем университетском кабинете в надежде на то, что студенты уже не будут отвлекать меня от работы. К моему несчастью, в расположенной рядом по коридору исследовательской лаборатории по машинной графике почти постоянно задерживались студенты, проводившие вечера за своей излюбленной игрой, называвшейся тогда по-разному: то «Звездный бой», то «Космическая война». «Стреляй! Скорее! Попал, попал! — то и дело доносились возгласы увлеченных игроков через плотно закрытую дверь моего кабинета. — Берегись, ракеты!» Этот шум не давал мне сосредоточиться. И, оказавшись не в состоянии работать, я обычно присоединялся к играющим. Или иногда, полный твердой решимости вернуться к более серьезным занятиям после небольшого развлечения, я отправлялся в соседнюю лабораторию понаблюдать за игрой.

Игра «Звездный бой» в какой-то мере была основана на сюжете популярного многосерийного телевизионного фильма под таким же названием. Игра повторяет эпизоды боя между космическим кораблем «Энтерпрайз» и крейсером вооруженных сил клингов. Поклонники этой телевизионной программы, наверное, помнят, что «Энтерпрайз» совершил путешествие в ту область Вселенной, где люди еще никогда не бывали. Это путешествие совершалось под флагом союза сотрудничавших цивилизаций, называемого Федерацией. Главенствующую роль в союзе играла человеческая раса. Клинги были представителями звероподобной волосатой расы, соперничавшей с Федерацией в борьбе за господство над Млечным Путем, а может быть, и над всей Вселенной.

Хотя первоначально игра «Звездный бой» могла выполняться лишь на мощном компьютере, оснащенном специальными графическими средствами, сейчас аналогичную программу вполне можно написать и для персонального компьютера. Два космических корабля описывают орбитальные траектории вокруг звезды, находящейся в центре, стреляя друг по другу ракетами и пытаясь в свою очередь уклониться от ракет противника. Как сами корабли, так и ракеты, которыми они обстреливают друг друга, подвержены гравитационному притяжению, и, таким образом, законы орбитального движения в значительной мере определяют характер игры. Пилоты, не обладающие достаточным интуитивным чутьем к законам небесной механики, либо сжигают свои корабли, неосторожно приблизившись к Солнцу, либо, не сумев сманеврировать, пересекают обманчивые траектории смертоносных ракет. Сторона, космический корабль которой взрывается первым, считается проигравшей.

Было время, когда различные версии игры «Звездный бой» распространились на сотни университетов и других учреждений. Официально это не одобрялось, но втайне все получали от игры много удовольствия. Хотя эта игра так часто мешала мне работать, теперь, через много лет я вспоминаю о ней с очень теплым чувством. «Звездный бой» — это лишь одна из многих увлекательных игр, созданных студентами. Большинство таких игр быстро превращались в законченные пакеты программ, представлявшие собой уже коммерческий продукт. По-видимому, игры внесли значительный вклад в ту революцию, которая произошла в области домашних компьютеров. Версии игры «Звездный бой» поступили в широкую продажу лишь сравнительно недавно, хотя в «подпольном» виде они кочевали уже давно.

Версия «Звездного боя», описанная в настоящей статье, вернет читателя к тем временам, когда появились первые компьютерные игры, окруженные атмосферой таинства и романтики. На ее примере я смогу посвятить читателя в секреты, без которых не обойтись при программировании игр. Прежде всего нужно стараться поддерживать непрерывное движение на экране. Поэтому вычисления в программе, реализующей игру «Звездный бой», должны быть как можно проще и выполняться как можно быстрее. С этой проблемой мы уже сталкивались в статье, посвященной программам, имитирующим полет (см.: «В мире науки», 1986, № 9, с. 92). К счастью, мир «Звездного боя» намного проще той детальной цифровой географии, которая должна раскрываться перед пилотом, сидящим за клавиатурой.

Вторая цель, стоящая перед программистом, заключается в том, чтобы создать реалистическую гравитационную среду. С этой проблемой читатели также уже сталкивались, правда, в несколько другой форме. В мартовском номере журнала рассказывалось об искусственной вселенной, за эволюцией которой можно наблюдать, сидя в кресле перед своим домашним компьютером. Звезды, населяющие этот мир, кружат причудливый хоровод, движимые силами их взаимного притяжения. Траектории кораблей и ракет в игре «Звездный бой» управляются более простым гравитационным влиянием. Лишь Солнце, находящееся в центре, оказывает ощутимое гравитационное воздействие.

Чтобы вступить в орбитальную дуэль, двое играющих усаживаются перед клавиатурой компьютера и нажимают на клавиши, специально отведенные в программе для управления кораблями. Один из играющих (по-видимому, более волосатый) принимает командование кораблем клингов, другому вверяется судьба корабля Федерации. Вначале на экране видны лишь Солнце, расположенное в центре, и два космических корабля. Солнце изображено в виде маленького диска, а корабли — крошечными символами, размеры которых достаточны лишь для того, чтобы отличить «своего» от «чужого».

В начале игры оба корабля находятся в состоянии свободного падения по направлению к Солнцу. Игроки немедленно меняют курс, стремясь уйти с траектории, ведущей к светилу, и включают ракетные двигатели кораблей, чтобы вывести их на безопасные орбиты. Корабль, коснувшийся Солнца, немедленно испаряется, и игра, конечно, оказывается проигранной.

Как только корабли выходят на заданные орбиты, каждый игрок начинает попытки уничтожить своего противника. Один из крайних тактических замыслов заключается в том, чтобы выждать, пока корабль противника не окажется поблизости. Тогда быстрым залпом его можно уничтожить. Другой крайний замысел, пожалуй, похитрее: можно попытаться произвести прицельные выстрелы с противоположной от Солнца стороны. На экране от ведущего огонь корабля отделяются яркие световые точки (в телевизионной программе называвшиеся фотонными

торпедами) и прожигают траектории вокруг Солнца в виде плавно закругляющихся смертоносных линий. Если кораблем противника управляет пилот, не обладающий высочайшим мастерством, корабль наверняка попадет под удар одной из выпущенных по нему ракет и, взорвавшись, разлетится на осколки, подобные звездной пыли. На экране это событие мгновенно знаменуется облаком светящихся точек и крупной надписью: ПОБЕДУ ОДЕРЖАЛА ФЕДЕРАЦИЯ или ПОБЕДИЛИ ВООРУЖЕННЫЕ СИЛЫ КЛИНГОВ.

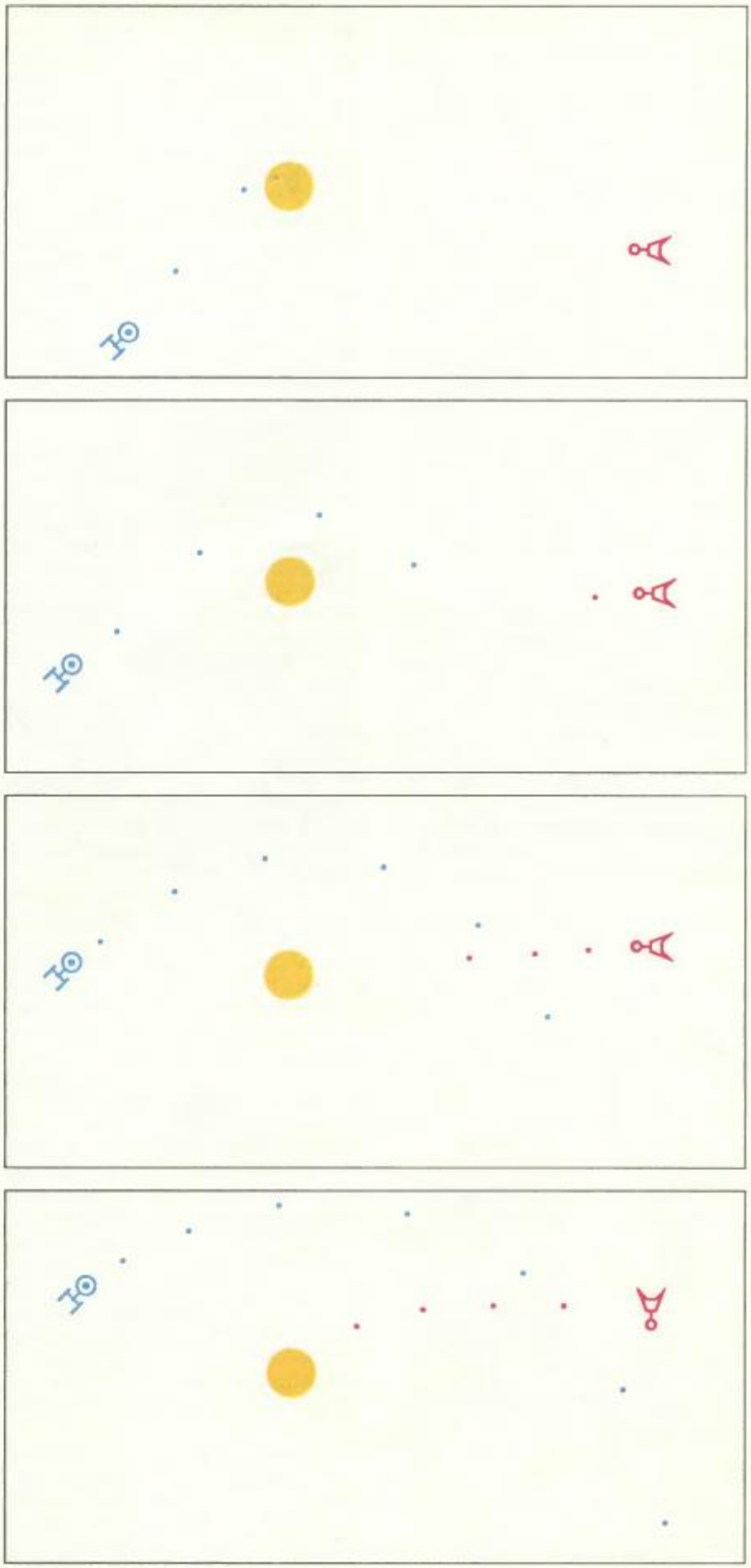
Поскольку боевые действия вблизи Солнца обычно очень интенсивны, осторожные игроки предпочитают держаться на более далеких орбитах. Основной недостаток подобной стратегии состоит в том, что возникает потребность в перезарядке солнечных элементов питания (системы управления кораблем питаются от солнечных батарей). Когда запасы электроэнергии исчерпываются, корабль должен немедленно приблизиться к Солнцу и пополнить энергетические запасы. На удаленных орбитах фотонный поток слаб и корабль подвергается риску потерять управление.

В игре есть еще одна тонкость, о которой следует помнить, находясь на удаленных от Солнца орбитах. Дело в том, что боевое пространство «Звездного боя» как бы «тороидально». Другими словами, если корабль подходит слишком близко к краю экрана, он исчезает и появляется на противоположной его стороне.

Каждый корабль обладает неограниченным запасом ракет, но в каждый данный момент в полете может находиться не более 10 из них. Ракета подчиняется тем же физическим законам, что и корабли. Она остается активной до тех пор, пока не поразит корабль (возможно, даже и свой), либо пока у нее не кончится горючее. Таковы правила игры.

Программа, которую я назову TREK (боевой марш), будет, наверное, самой крупной из тех, что когда-либо рассматривались в статьях рубрики «Занимательный компьютер», и, возможно, не столько из-за своей сложности, сколько из-за масштабов. Поэтому некоторые более или менее стандартные процедуры мы рассмотрим лишь кратко, в общих чертах. Тем не менее программисты-новички, в особенности те, у кого есть возможность проконсультироваться у опытных специалистов, могут попытаться написать собственную программу с надеждой на успех.

Программа TREK циклически проходит через шесть основных секций до тех пор, пока оба корабля сущест-



«Энтерпрайз» и крейсер клингов обстреливают друг друга ракетами

вуют:
Опрос клавиш
Корректировка положения кораблей и ракет
Проверка на столкновения
Изменение энергетических ресурсов кораблей
Управление ракетами
Вывод изображения на экран

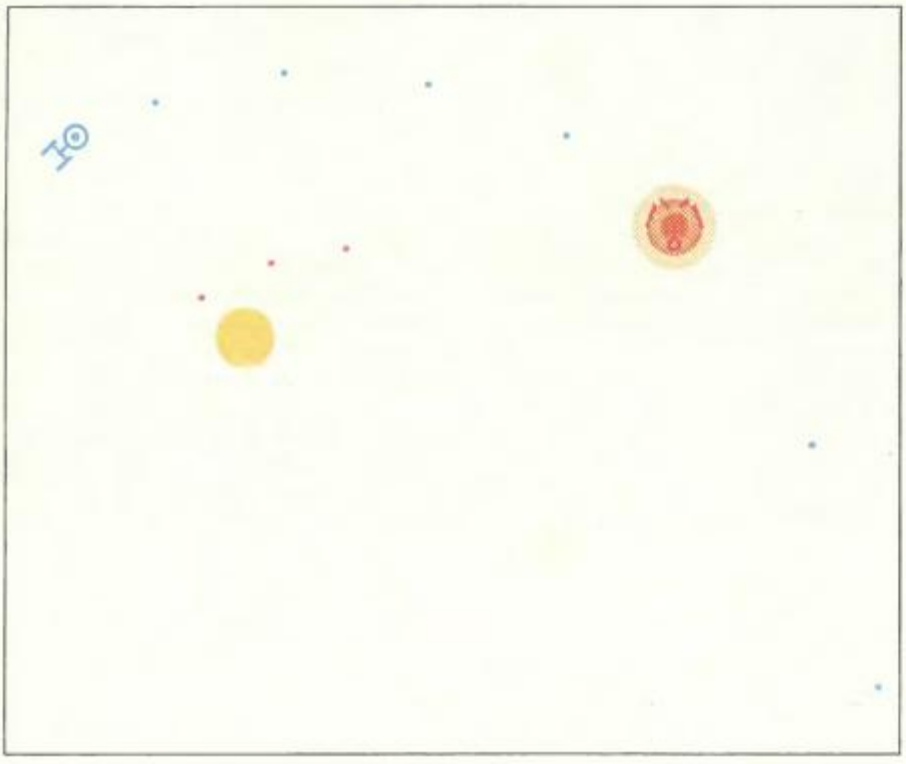
Многие читатели найдут программу TREK сравнительно несложной, может быть, за исключением первой секции. Для некоторых такая процедура, как опрос клавиш, будет в новинку, однако при программировании игр без нее, пожалуй, не обойтись. В большинстве языков высокого уровня имеются специальные операторы, позволяющие проверить, была ли нажата та или иная клавиша.

Например, в языке Бейсик версии Microsoft соответствующий оператор выглядит следующим образом:

On key (*k*) gosub *n*.

После того как выполняется команда On key, программа, перед тем как выполнять каждый следующий оператор, проверяет, не была ли нажата клавиша *k*. Если была, то управление в программе переходит на первый оператор в строке под номером *n*. В этой строке начинается подпрограмма, назначение которой заключается в том, чтобы зафиксировать каким-то образом тот факт, что клавиша *k* была нажата. Обычно это делается путем присвоения определенного значения переменной, играющей роль «флажка». На клавиатуре персонального компьютера IBM имеется 14 клавиш, которые можно опросить указанным способом: 10 функциональных клавиш и четыре клавиши управления курсором. Каждому играющему нужно отвести по четыре клавиши, скажем функциональные клавиши *F*1, *F*2, *F*3 и *F*4 для корабля Федерации и четыре клавиши управления курсором для корабля клингов. Числа, соответствующие первым четырем клавишам, — от 1 до 4, а другим четырем — от 11 до 14. Здесь нужно посмотреть коды клавиш в руководстве по программированию на том или ином компьютере.

Для каждой стороны первая из выделенных ей клавиш управляет мощностью двигателя, две следующие служат для изменения направления и последняя — для стрельбы ракетами. Клавиша двигателя сообщает кораблю фиксированный импульс мощности, действующий в течение одного программного цикла. Каждое нажатие на клавишу направления поворачивает соответствующий корабль на угол, равный 10°. При нажатии на клавишу стрельбы корабль выпускает один снаряд. Прежде чем начать игру, советуем налепить на клавиши маленькие квадратики с обозначениями их функций. Они будут напоминать играющим назначение каждой клавиши.



Корабль клингов не смог уйти от огня

В программе должно быть по одному оператору On key для каждой из восьми клавиш управления. Оператор On key не выполняется, если ему не предшествует оператор Key(*k*)on. Все процедуры, к которым обращается программа по каждой команде On key, очень просты и в сущности одинаковы. Каждая такая процедура состоит из двух операторов. Первый оператор присваивает переменной-флажку значение 1 для последующего анализа в ходе выполнения программы, а второй оператор возвращает управление на то место в программе, откуда произошло обращение к данной процедуре по команде On key. Переменные-флажки для управления «Энтерпрайзом», можно, например, назвать *fdgo, fdrt, fdlt* и *fdfr*. Эти названия означают соответственно «Федерация вперед» (Federation go) — включение двигателя, «Федерация вправо» (Federation right), «Федерация влево» (Federation left) — эти повороты делаются соответственно по и против часовой стрелки — и «Федерация огонь» (Fedaration fire) — старт боевой ракеты. Аналогично для крейсера клингов переменные *kngo, knrt, knlt* и *knfr* соответственно регистрируют нажатие клавиш импульса, направления и стрельбы.

После того как процедура присваивает переменной-флажку значение 1, в состоянии одного из кораблей происходит изменение. Например, в случае если *fdgo* становится равной 1, секция программы, управляющая положением кораблей, создает небольшое дополнительное ускорение (в пределах между 2 и 5, кому как нравится) в движении корабля «Энтерпрайз». Затем программа должна сбросить флажок в 0.

Управлять положением двух космических кораблей и несколькими ракетами значительно проще, чем смоделировать движение такого же числа массивных звезд. Суммарная масса боевой техники пренебрежимо мала по сравнению с массой расположенного в центре системы Солнца, поэтому взаимным гравитационным притяжением, действующим между кораблями и ракетами, мы пренебрегаем. По заданному расстоянию от каждого объекта до Солнца программа TREK вычисляет ускорение этого объекта, обусловленное радиально направленной силой притяжения, по этому ускорению вычисляется изменение скорости объекта и, наконец, его перемещение.

Даже при таком в принципе простом характере вычислений они отнимают значительное количество машинного времени. Для выполнения этих вычислений требуются операции

сложения, умножения и извлечения квадратного корня. Чтобы избежать замедления темпа, в котором проводится игра, в программе TREK используется таблица: для каждого возможного расстояния от объекта до Солнца массив под названием *force* (сила) указывает заранее вычисленное ускорение, которое должен испытывать объект (см. рисунок вверху справа). Поскольку мир этой игры «тороидален», новое положение объекта, вычисляемое по величине ускорения в массиве *force*, должно определяться по модулю, равному максимальному горизонтальному или вертикальному расстоянию, представленному на прямоугольном экране дисплея.

В двух других массивах содержатся данные о скорости и положении обоих космических кораблей, а также до 20 выпущенных ими ракет. Эти массивы называются соответственно *vel* (от velocity — скорость) и *pos* (от position — положение). Каждый массив содержит два стобца и 22 строки. В первых двух строках находятся данные о космических кораблях, а в остальных 20 — о ракетах. Так, например, в первой строке массива *vel* два элемента *vel* (1,1) и *vel* (1,2) указывают соответственно составляющие скорости по горизонтальной оси *x* и по вертикальной оси *y* для корабля «Энтерпрайз». Аналогично, элементы *vel* (2,1) и *vel* (2,2) задают две взаимно перпендикулярные компоненты скорости для крейсера клингов. Специальная переменная *misnum* (от missile — ракета и number — число) позволяет программе следить за количеством ракет, активных в каждый данный момент. Значение *misnum* изменяется в пределах от 2 (т. е. ни одной ракеты, только сами корабли) до 22 (т. е. 2 корабля и 20 боевых ракет).

Разместив все данные о кораблях и снарядах в единых массивах, мы создаем более короткую и более быструю программу. Для вычисления новых скоростей и положений требуется лишь один цикл. Однако для определения изменений скорости и положения кораблей требуются еще две переменные *fdor* и *knor*. Их значения выражены в градусах, причем если угол равен 0, то корабль смотрит строго на восток, если угол равен 90, то корабль смотрит на север, и т. д. Всякий раз, когда игрок нажимает клавишу направления, значение одной из этих переменных изменяется на 10° в ту или другую сторону.

Цикл для определения нового положения объекта проиндексирован переменной *i*, принимающей значения от 1 до 22. Для каждого значения *i* программа вычисляет ускорение: текущие значения *x* и *y*, указанные в *i*-й строке массива *pos*, возводятся в квадрат и складываются. Затем из полученной суммы программа извлекает квадратный корень и результат усекается до ближайшего целого числа, меньшего или равного этому результату. Усеченная таким образом величина представляет собой приблизительно расстояние от Солнца до точки, в которой находится объект с индексом *i*. Это целое число используется затем в качестве индекса в таблице ускорений, по которой программа определяет силу притяжения к Солнцу.

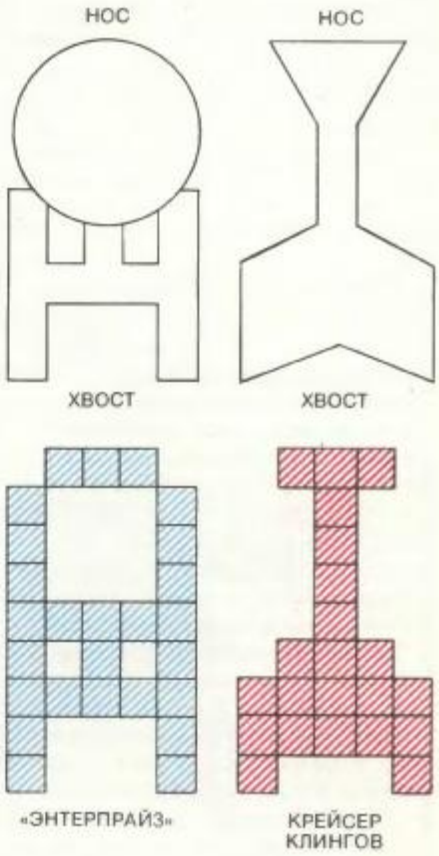
Для каждого из двух космических кораблей к компонентам ускорения, обусловленного притяжением к Солнцу, нужно прибавить компоненты ускорения, получаемого в результате импульсов двигателя. Если значение какого-либо флажка, либо *fdgo*, либо *kngo*, установлено в 1 (другими словами, если была нажата клавиша двигателя) и соответствующий корабль еще не исчерпал запасы топлива, программа должна умножить постоянную импульса (его абсолютную величину) на горизонтальную и вертикальную компоненты. На своем компьютере я выбрал константу импульса равной 3, что обеспечивает космическим кораблям достаточно хорошую маневренность. Для «Энтерпрайза» горизонтальная компонента равна утроенному значению косинуса *fdor*, а вертикальная — утроенному значению синуса *fdor*. Для корабля клингов компоненты импульса вычисляются аналогично по углу *knor*. После того как эти вычисления завершены, программа снова устанавливает значения флажков *fdgo* и *kngo* равными 0, ускорители заглушаются до тех пор, пока вновь не будет нажата одна из клавиш, включающих двигатель.

Когда индекс *i* становится больше 2, процедура расчета импульса пропускается, поскольку однажды выпущенные по цели боевые ракеты уже находятся в пассивном полете. В оставшейся части цикла корректировки вычисляются новые скорости и положения движущихся на экране объектов. Для каждого объекта численное значение ускорения прибавляется к текущему значению скорости, а величина скорости — к соответствующей координате положения. Такое упрощение в вычислениях достигнуто за счет соответствующего выбора величины одиночного импульса и солнечного притяжения, при котором минимальный промежуток времени в основном программном цикле оказывается равным единице.

Чтобы провести проверку на контакт между различными движущимися объектами, программа прежде всего должна установить, находятся ли корабли вне границ солнечного диска. Поскольку расстояние между каждым кораблем и Солнцем уже было вычислено, то остается лишь прове-

РАССТОЯНИЕ	СИЛА
10	8,000
11	6,612
12	5,556
. . .	
$\text{СИЛА} = \frac{800}{(\text{РАССТОЯНИЕ})^2}$	
. . .	
178	0,025
179	0,025
180	0,025

Часть таблицы ускорения и его формула



Символы кораблей (вверху) *и их изображения на экране*

рить, превышает ли это расстояние радиус Солнца, скажем, 10 единиц. Если один из кораблей коснулся светила, программа выводит на экран соответствующее сообщение, например: КЛИНГИ ИСПАРИЛИСЬ. Затем управление передается секции изображения на экране очередной картинки.

В ходе второй проверки на контакт нужно определить, не находится ли какой-нибудь снаряд в пределах порогового маленького расстояния до одного из космических кораблей. Здесь используется простой, но эффективный прием: сначала вычисляется разность между значениями координат снаряда и корабля по оси *x*, затем по оси *y*, после чего эти две разности складываются, и таким образом мы избегаем операций возведения в квадрат и извлечения квадратного корня; результат почти не отличается от вычисленного обычным способом. Если сумма двух разностей оказывается меньше,скажем, 4, то программа фиксирует попадание. На экране появляется сообщение, например такое: ЭНТЕРПРАЙЗ ПОРАЖЕН РАКЕТОЙ. КЛИНГИ ОДЕРЖАЛИ ПОБЕДУ. Проверка для всех выпущенных на данный момент ракет производится в одном цикле. Его индекс пробегает значения от 3 до *misnum*, т. е. до числа ракет плюс 2.

Чтобы изменить запасы энергии, которыми располагают корабли, программа делит значение ускорения, обусловленного притяжением к Солнцу и получаемого из таблицы, на 60. Поскольку величина ускорения растет при приближении к Солнцу, корабль получает больше энергии благодаря более интенсивному потоку солнечных фотонов. Получаемая энергия прибавляется затем к энергетической переменной *fdfl* или *knfl* в зависимости от того, для какого корабля проводятся вычисления. Каждая из этих переменных уменьшается на 0,1 при нажатии на клавишу двигателя. Это уменьшение производится в секции корректировки положения. Считается, что корабль исчерпал свои запасы энергии, когда в его баке остается меньше одной единицы солнечного топлива. Вначале в баке каждого корабля имеется по 10 единиц топлива.

Управление боевыми ракетами осуществляется с помощью специального массива *time* (время), хранящего число программных циклов, в которых выражается «возраст» каждой ракеты. Когда число циклов достигает 25, соответствующая ракета выводится из массива *pos*, счетчик сбрасывается на 0 и значение переменной *misnum* уменьшается на 1. Удалить ракету из *pos* можно двумя способами. Первый способ легче реализовать в программе, но он замедляет темп игры. В этом случае программа пробегает массив, начиная со значения индекса, соответствующего удаляемой ракете, и уменьшает номер строки каждого элемента на 1. Таким образом, последний передвигаемый элемент имеет индекс *misnum*. Он перемещается на строку с номером *misnum* − 1. Та же самая операция выполняется над массивами *vel* и *time*.

Более быстрый способ основан на учете индекса ракеты, который тем меньше, чем старше ракета, поскольку она была раньше добавлена к списку. Поэтому можно следить за ракетами, не передвигая их индексов. Ракеты, возраст которых достиг 25 программных циклов, должны находиться в начале последовательно расположенных ракет в каждом массиве, и только они подлежат удалению. Аналогичным образом, новые ракеты всегда добавляются в конец последовательности.

Введем две новые переменные *old* (старый) и *new* (новый), которые будут служить в качестве указателей на самую старую и самую новую ракету в каждом массиве. По мере того как одни снаряды удаляются, а другие добавляются к спискам, изменению подвергаются лишь значения *old* и *new*. В таком случае можно воспользоваться арифметическими действиями, выполняемыми по модулю, и тем самым поддерживать циклическое перемещение ракет по массивам. Когда нужно добавить новую ракету при значении индекса, равном 23, программа берет индекс по модулю 23, получает 0 и прибавляет к нему число 3, чтобы не подменить координаты космического корабля координатами одной из ракет. Значения *old* и *new* подвергаются той же самой операции. Такая структура данных называется круговой очередью. Если мы решили воспользоваться этим приемом, то секция программы, вычисляющая коррекции положения, должна быть модифицирована: каждый цикл нужно разбить на два меньших цикла, один для кораблей и другой для ракет.

Когда играющий нажимает клавишу стрельбы, программа сначала проверяет, сколько снарядов активны на данный момент у данной стороны. Если их насчитывается меньше 10, то анализируются значения переменных-флажков *fdfr* и *knfr*. Если, например, *fdfr* равна 1, то программа прибавляет 1 к количеству ракет на стороне Федерации, увеличивает на 1 значение *misnum* и затем помещает координаты положения и скорости корабля «Энтерпрайз» в соответствующие элементы массивов *pos* и *vel*. При этом к координатам положения прибавляется по 4 единицы и по 2 к координатам скорости. Эти увеличения происходят в том направлении, в котором корабль движется в данный момент. Например, горизонтальная координата ракеты, выпускаемой «Энтерпрайзом», равна учтверенному косинусу угла *fdor* плюс текущая горизонтальная координата положения корабля в данный момент; для вертикальной координаты мы берем соответственно учтверенное значение синуса *fdor*. Таким образом, сразу после запуска ракета уже достаточно удалена от своего корабля, иначе он был бы мгновенно уничтожен своей собственной фотонной торпедой. Аналогичная процедура применяется к координатам скорости снаряда. Это означает, что в момент запуска скорость ракеты относительно корабля составляет две единицы на один временной цикл. Другими словами, скорость снаряда на две единицы больше, чем у запустившего его корабля.

Последняя из числа основных секций программы TREK изображает на экране Солнце, два корабля и все активные к данному моменту ракеты. Программа рисует круг радиусом 10 единиц в центральной части экрана и затем просматривает массив *pos*. Корабли Федерации и клингов представлены маленькими символическими изображениями. Одно из этих изображений по существу представляет собой кружок (в фильме «Энтерпрайз» имел форму диска с двумя слегка выступающими двигателями). Корабль клингов имеет угловатую форму. Читатели могут сами выбрать символы для представления на экране кораблей, учитывая, чтобы принадлежность корабля и направление его движения можно было определить с первого взгляда. При изображении на экране символов обоих кораблей программа TREK пользуется специальными списками точек, по которым строится изображение и которые можно переносить и подвергать поворотам, моделируя тем самым изменения в положении и ориентации кораблей. Для выполнения этих операций программа обращается к массиву *pos* и переменным *fdor* и *knor*.

С ракетами дело обстоит проще: каждая изображается на экране в виде точки. Это происходит в ходе выполнения одного цикла при последовательном просмотре массива *pos*.

Использование программы TREK сопряжено с некоторым неудобством, возникающим в связи с конструктивными особенностями большинства переменных дисплеев. Дело в том, что экраны этих дисплеев работают в режиме запоминания изобра-

жения: объект, однажды «нарисованный» на экране, остается на нем продолжительное время. Чтобы избежать загромождения экрана «остатками» изображений кораблей и ракет из предшествующих стадий игры, программа должна рисовать каждый объект дважды. Сначала она рисует его на старой позиции черным, а затем изображает его на новой позиции светлым.

Остальные подробности реализации игры я оставляю ее будущим энтузиастам, «треккерам». Несмотря на некоторые хитрости, при помощи которых удается ускорить программу, вы все же можете остаться неудовлетворенными темпом игры. Возможно, по этой причине вы назовете свою программу не TREK, а TRAKTOR. Чтобы попытаться ускорить работу программы, попробуйте ее откомпилировать или наложите более жесткие ограничения на число активных боевых ракет, разрешенных для каждой стороны.

Я описал здесь лишь основную, так сказать, скелетную версию игры «Звездный бой». Существуют и более замысловатые версии, их число растет и они также получают распространение. Появились варианты игры с тремя и более кораблями, лазерными пушками, цветным изображением и высветкой текущих показателей игры. Я должен выразить признательность Дж. Гроффу из Клирватера (шт. Флорида), напомнившему мне об этой классической игре, родившейся в подполье, и познакомившему меня с версией игры, в которой поведение клингов автоматизировано. Люди, пытающиеся отстаивать честь Федерации, все время терпят поражение в борьбе с программой Гроффа.

Ну что же, я предлагаю читателям, которым удастся создать собственные работающие версии «Звездного боя», рассказать в своих письмах о стратегиях и остроумных решениях, благодаря которым они добивались ускорения действия. О лучших результатах я расскажу в одной из следующих статей.

Абсолютное большинство писем, присланных читателями в ответ на статью о компьютерной магии (см.: «В мире науки», 1986, № 10), содержали ответы на вопрос, поставленный в этой статье, но объяснение было опубликовано уже в следующем номере. Вопрос был о том, каким образом факир, располагая лишь карманным калькулятором, мог «прочесть» мысли человека? Объяснение заключалось в том, что любое число, составленное из цифр, расположенных на одной из линий клавиатуры калькулятора, делится на 3. Я не буду еще раз подробно останавливаться на этом трюке, но хотел бы обратить внимание читателей на то, что обычно в статьях нашей рубрики ставятся вопросы двух типов и между ними существует одно принципиальное различие.

Вопросы первого типа требуют небольшого размышления. Ответы на такие вопросы публикуются в следующем номере журнала. При постановке вопроса автор предупреждает читателей, что ответ будет опубликован задолго до того, как придут первые отклики.

Вопросы второго типа требуют более солидной проработки. Как правило, чтобы ответить на них, необходимо воспользоваться компьютером, и при постановке таких вопросов автор просит читателя присылать ответы. На размышления предоставляется достаточно много времени. Если письмо выслано через две-три недели после публикации статьи, то у него имеются хорошие шансы попасть на мой стол, прежде чем обзор читательских писем отправится в печать. Обычно такой обзор появляется через три месяца после выхода статьи.

К. Фалвес из Тинека (шт. Нью-Джерси) предложил еще один вариант трюка с калькулятором. Как и прежде, факир просит добровольца из публики выбрать два числа из цифр, лежащих на одной линии клавиатуры калькулятора, и перемножить их, воспользовавшись калькулятором. В предложенном мною варианте доброволец не должен был называть одну из цифр произведения, а факиру нужно было «угадать» ее путем выбрасывания девяток. Несколько более занятным выглядит этот фокус, если доброволец из публики называет все цифры, но при этом добавляет возраст своей собачки или кошечки. Если возраст животного не превышает 9 лет, то факир может легко его подсчитать: «Мадам, ду́хи поведали мне, что вашему Рексику пять лет».

Р. Малленикс, инженер, работающий в фирме Georgia Power Company в Валдосте (шт. Джорджия), рассказал мне о своем коротком, но поучительном приключении с программой PREDICTABIT, которая должна предсказывать следующий бит, называемый человеком. После сравнительно длинной цепочки нулей и единиц частота правильных предсказаний должна повышаться. Это объясняется тем, что в подсознании человека, пытающегося выбирать цифры совершенно случайно, все же существуют некоторые уловимые закономерности. Малленикс добился сравнительно скромного успеха, играя с приятелем. Программа показывала результаты, точность которых была близка к 60%. Однако, когда приятель понял, как работает программа, он начал называть цифры, пользуясь разложением числа π. Число случаев правильного предсказания снизилось до 43%. Ну что же, я предупреждал читателей. Клиентов следует выбирать в среде доверчивых людей, кроме того, их нужно постоянно отвлекать какой-нибудь болтовней так, чтобы они не успевали задуматься.

Издательство
МИР
предлагает:

Дж. Уэстон
ТЕХНИКА СВЕРХВЫСОКОГО ВАКУУМА

Перевод с английского

В книге специалиста из Великобритании рассмотрены вопросы теории и практики сверхвысокого вакуума. Представлены сведения о материалах, используемых в условиях вакуума, средствах получения и сохранения высоких степеней разряжения и способах измерения давлений. Описаны методы контроля герметичности. Рассмотрены некоторые модели новейшего научного и промышленного оборудования.

Для специалистов в области вакуумной техники, а также для широкого круга экспериментаторов, работающих в смежных областях техники и использующих вакуум в своей практической деятельности.

Из рецензии академика А. И. Шальникова: «Существенной особенностью книги Дж. Уэстона является то, что в ней рассмотрены практически все аспекты проблемы получения ультравысокого вакуума, интересующее широкий круг читателей».

1988, 25 л. Цена 2 р. 20 к.

Издательство «Мир»